AMENDMENTS TO THE CLAIMS

Please amend only claims 1, 15 and 23. The remaining unamended claims are also provided below for convenience of reference.

1. **(currently amended)** A processor comprising:
a plurality of functional units; and
a substantially contiguous register file that is divided into a plurality of register file segments, ones of the plurality of register file segments being coupled to and associated with ones of the plurality of functional units, the register file segments being partitioned into global registers and local registers, the global registers that are accessible by the plurality of functional units, the local registers being accessible by the functional unit associated with the register file segment containing the local registers, wherein the number of global registers and the number of local registers are programmably configurable.

2. (original) A processor according to Claim 1 wherein:
the processor is a Very Long Instruction Word (VLIW) processor.

3. (original) A processor according to Claim 1 wherein:
the local registers and global registers are addressed using register addresses in an address space that is defined for a register file segment/ functional unit pair.

4. (original) A processor according to Claim 1 wherein:
the register file is a multi-ported register file.

5. (original) A processor according to Claim 1 wherein:
the local registers in a register file segment are addressed using register addresses in a local register range outside the global register range that are assigned within a single register file segment/ functional unit pair.

6. (original) A processor according to Claim I wherein:

register addresses in the local register range are the same for the plurality of register file segment/ functional unit pairs and address registers locally within a register file segment/ functional unit pair.

7. (previously amended) A processor according to Claim 1 wherein:

the register file includes M of the register file segments, with each of the M register file segments having N physical registers, the register file segments having a reduced number of read and/or write ports in comparison to an undivided register file.

8. (previously amended) A processor according to Claim 7 wherein:

the register file segments are partitioned into N_G global and N_L local register files where N_G plus N_L., is equal to N, the register file having $N_G + (M * N_L)$ total registers available for the M functional units, the number of address bits for addressing the $N_G + (M * N_L)$ total registers being equal to the number of bits B that are used to address $N = 2^B$ registers.

9. (previously amended) A processor according to Claim 8 wherein:

partitioning of the register file is programmable so that the number N_G of global registers and number N_L of local registers is selectable and variable.

10. (original) A processor according to Claim 1 wherein the register file is a storage array structure having R read ports and W write ports comprising:

a plurality of storage array storages;

the storage array storages having a reduced number of read ports so that the total number of read ports for the plurality of storage array storages is R read ports; and

the storage array storages having W write ports.

11. (previously amended) A processor according to Claim 10 wherein:

the storage array structure is a multi-port structure; and

the plurality of storage array storages includes four storage array storages each having three read ports and five write ports.

12. (previously amended) A processor according to Claim 10 wherein:
the storage array structure is a multi-port structure; and
the plurality of storage array storages includes four storage array storages each having three read ports and four write ports.

13. (previously amended) A processor according to Claim 10 wherein:
the writes for the global registers are fully broadcast so that all of the storage array storages are held coherent.

14. (previously amended) A processor according to Claim 10 wherein:
storage array storages include storage cells having a plurality of word lines and a plurality of bit lines, the word lines being formed in one metal layer, the bits lines being formed in a second metal layer.

15. **(currently amended)** A processor comprising:
a decoder for decoding a very long instruction word including a plurality of sub instructions, the sub instructions being allocated into positions of the instruction word;
a substantially contiguous register file coupled to the decoder and divided into a plurality of register file segments; and
a plurality of functional units, ones of the plurality of functional units being coupled to and associated with respective ones of the register file segments, ones of the plurality of sub instructions being executable upon respective ones of the plurality of functional units, operating upon operands accessible to the register file segment associated with the functional unit of the plurality of functional units, the register file segments including a plurality of registers that are partitioned into global registers and local registers, the global registers being accessible by the plurality of functional units, the local registers in one of the register file segments being accessible by the functional unit associated with the register file segment.

16. (original) A processor according to Claim 15 wherein:
the local registers and global registers are addressed using register addresses in an address space that is defined for a register file segment/ functional unit pair.

17. (original) A processor according to Claim 15 wherein: the register file is a multi-ported register file.

18. (original) A processor according to Claim 15 wherein:
the local registers in a register file segment are addressed using register addresses in a local register range outside the global register range that are assigned within a single register file segment/ functional unit pair.

19. (original) A processor according to Claim 15 wherein:
register addresses in the local register range are the same for the plurality of register file segment/ functional unit pairs and address registers locally within a register file segment/ functional unit pair.

20. (previously amended) A processor according to Claim 15 wherein:
the register file includes M of the register file segments, with each of the M register file segments having N physical registers, the register file segments having a reduced number of read and/or write ports in comparison to an undivided register file.

21. (previously amended) A processor according to Claim 20 wherein:
the register file segments are partitioned into N_G global and N_L local register files where N_G plus N_L is equal to N, the register file having $N_G + (M * N_L)$ total registers available for the M functional units, the number of address bits for addressing the $N_G + (M * N_L)$ total registers being equal to the number of bits B that are used to address $N = 2^B$ registers.

22. (previously amended) A processor according to Claim 21 wherein:
partitioning of the register file is programmable so that the number N_G of global registers and number N_L of local registers is selectable and variable.

23. **(currently amended)** A method of operating a processor, the processor including a plurality of functional units and a substantially contiguous register file divided into a plurality of register file segments, the plurality of register file segments being coupled and associated to ones of the plurality of functional units, comprising:

partitioning the register file segments into global registers and local registers;

operating the plurality of functional units;

accessing the global registers by the plurality of functional units;

accessing the local registers by the functional unit associated with the register file segment containing the local registers; and

programmably partitioning the register file so that the number of the global registers and the number of the local registers are selectable and variable.

24. (original) A method according to Claim 23 further comprising:

addressing the local registers and global registers using register addresses in an address space that is defined for a register file segment/ functional unit pair.

25. (original) A method according to Claim 23 further comprising:

addressing the local registers in a register file segment using register addresses in a local register range outside the global register range that are assigned within a single register file segment/ functional unit pair.

26. (original) A method according to Claim 23 further comprising:

addressing the local register range the same for the plurality of register file segment/ functional unit pairs and address registers locally within a register file segment/ functional unit pair.

27. (previously amended) A method according to Claim 23, wherein the register file include M of the register file segments, with each of the M register file segments having N physical registers, the register file segments having a reduced number of read and/or write ports in comparison to an undivided register file.

28. (previously amended) A method according to Claim 27 further comprising:

partitioning the register file segments into N_G global and N_L local register files where N_G plus N_L is equal to N; and

operating the register file having $N_G + (M * N_L)$ total registers available for the M functional units, the number of address bits for addressing the $N_G + (M * N_L)$ total registers being equal to the number of bits B that are used to address $N = 2^B$ registers.